

Figure F

FIGURE A. TI-99/4A SYSTEM BLOCK DIAGRAM

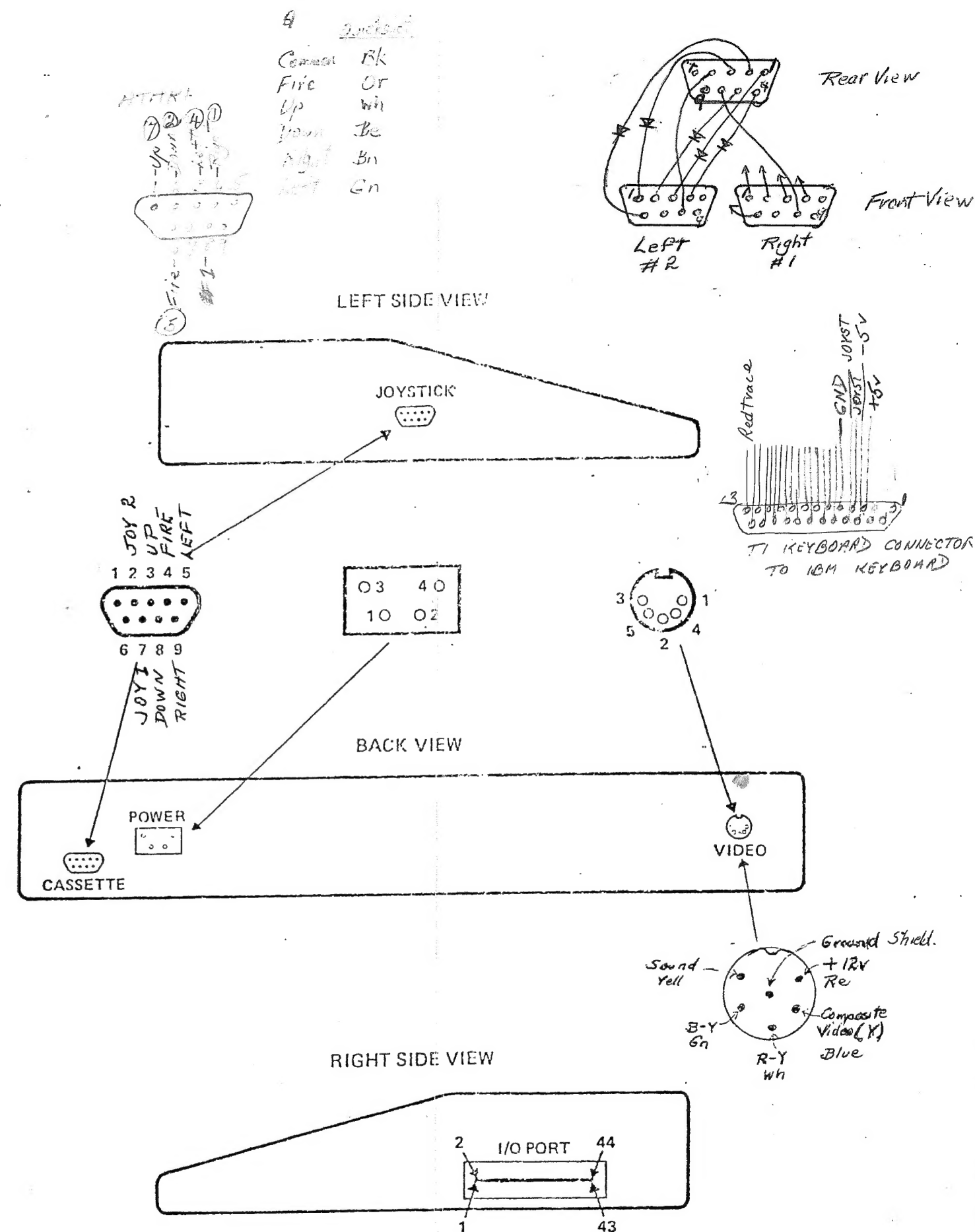
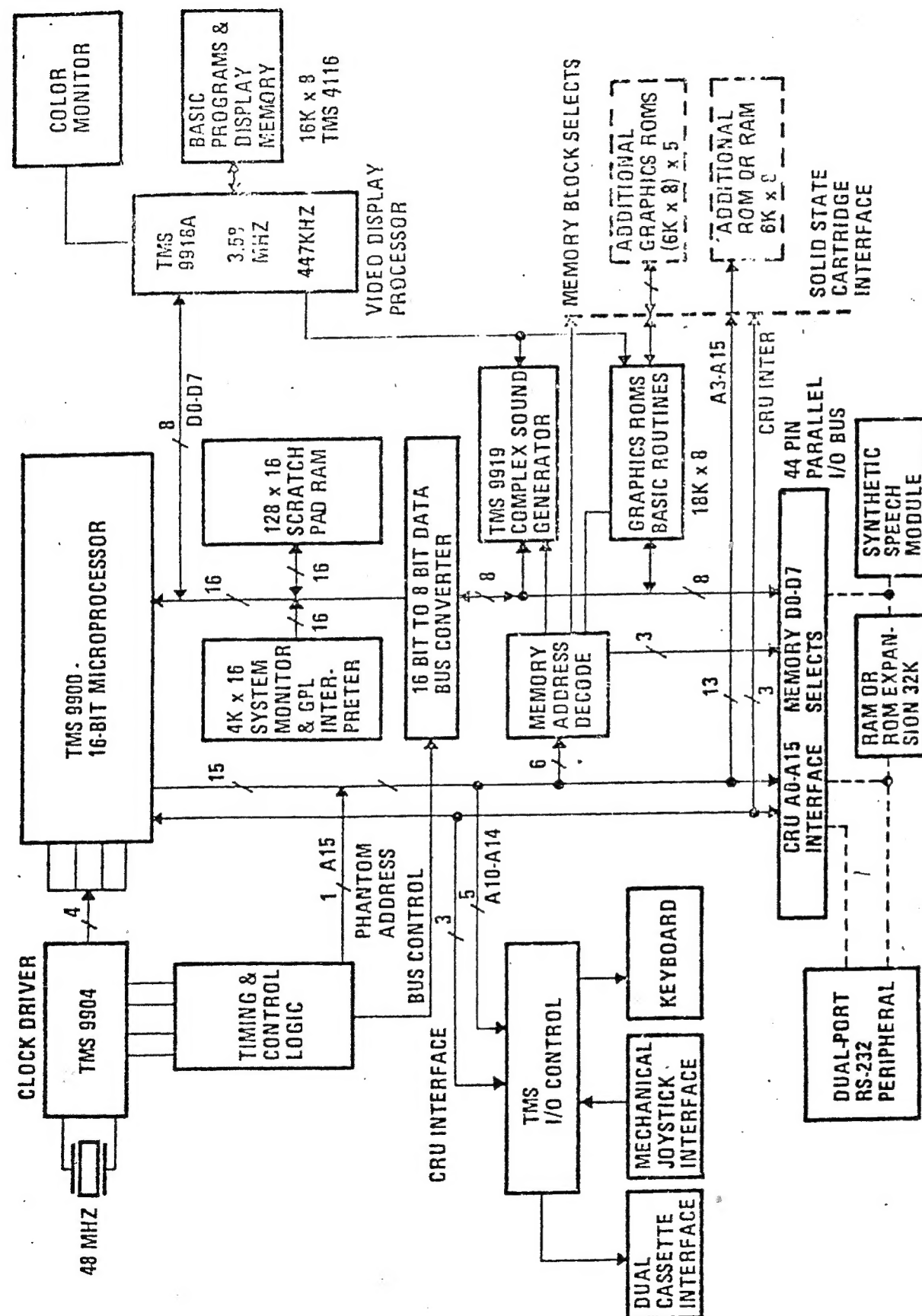
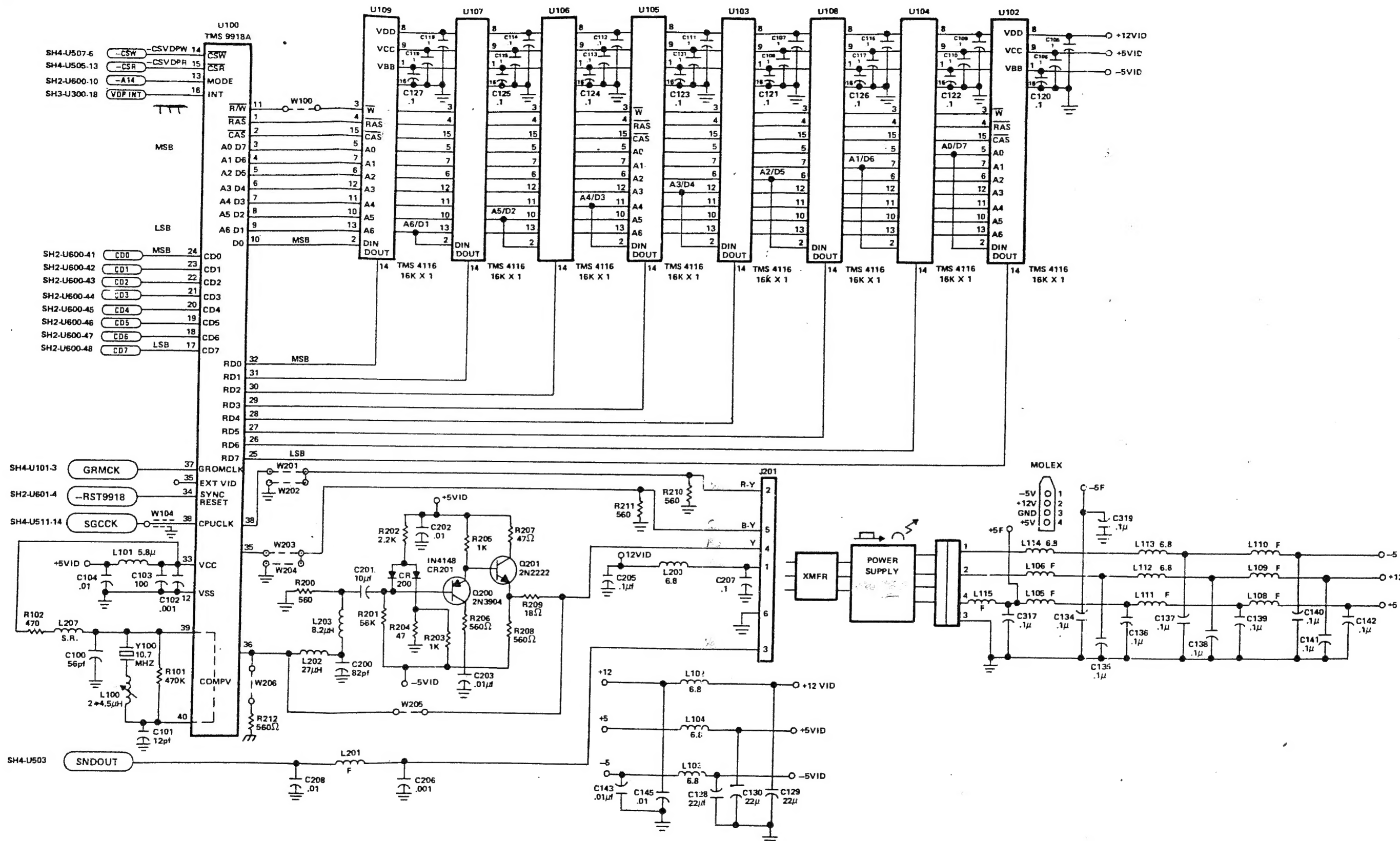
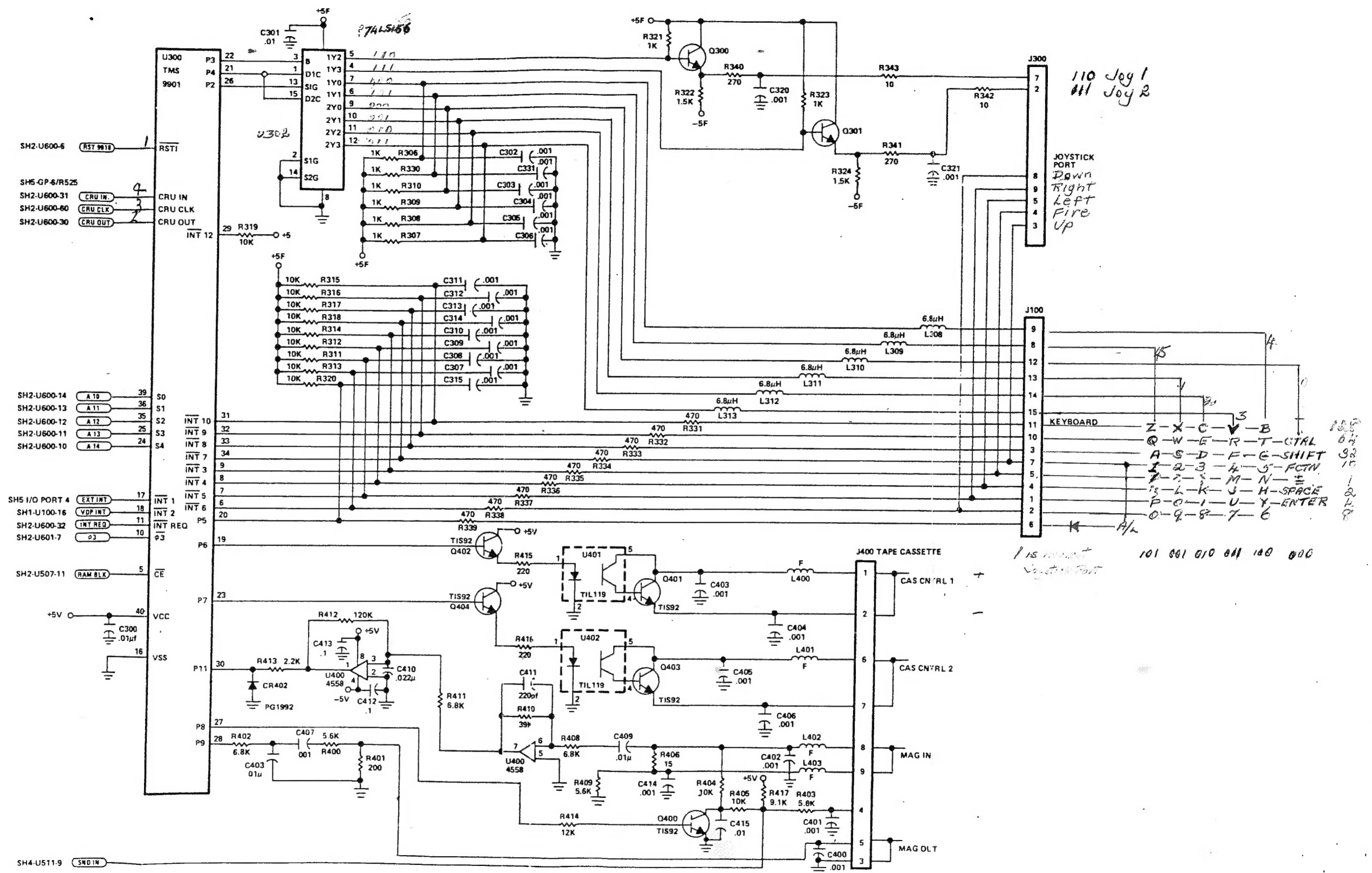


FIGURE E. CONNECTOR PIN IDENTIFICATION DIAGRAM

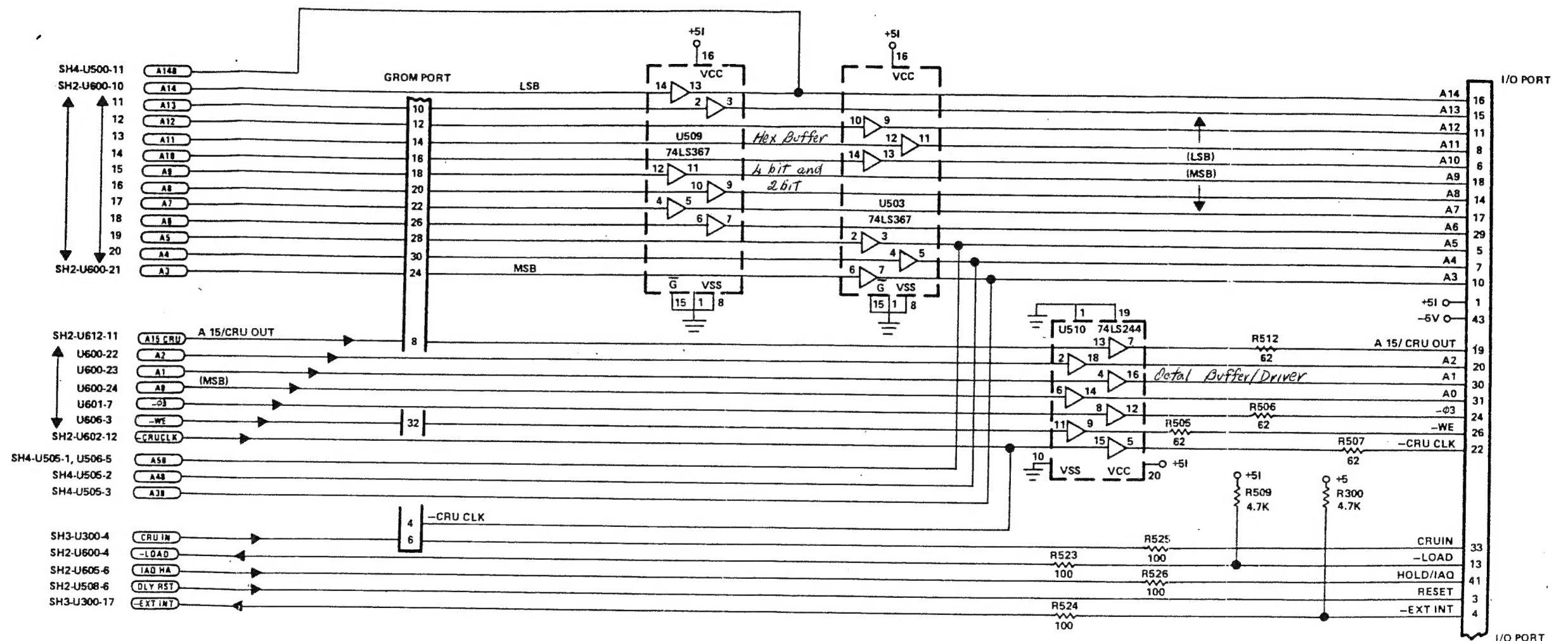


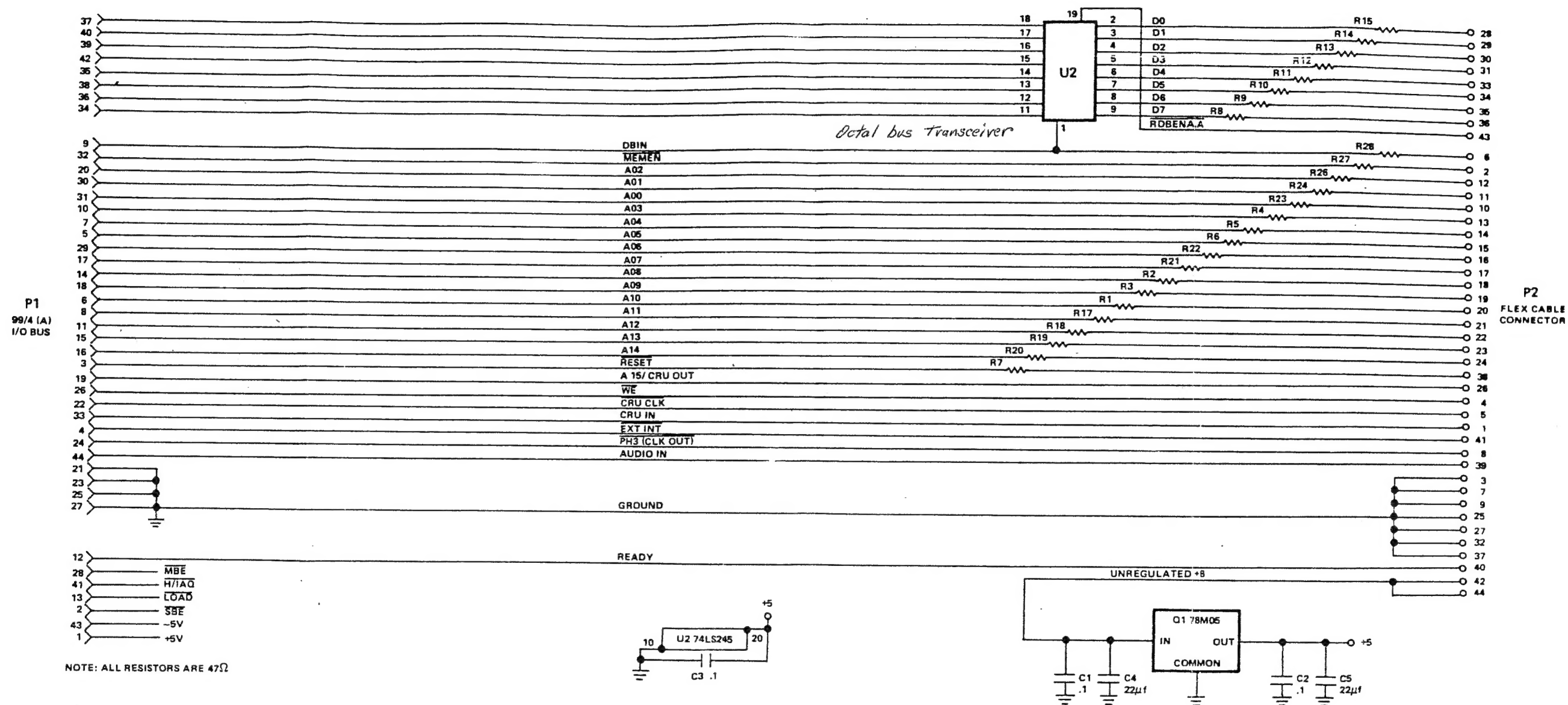




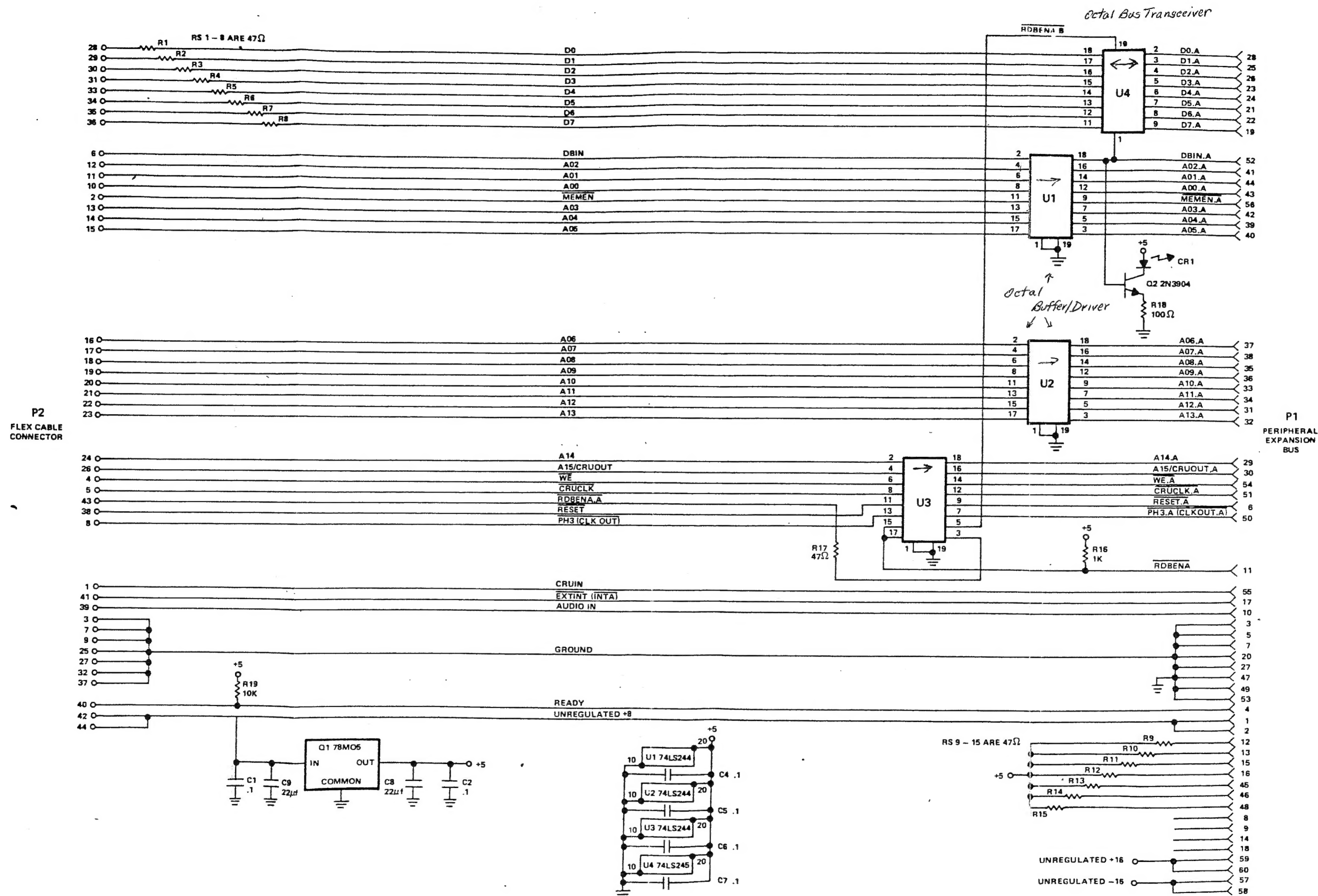




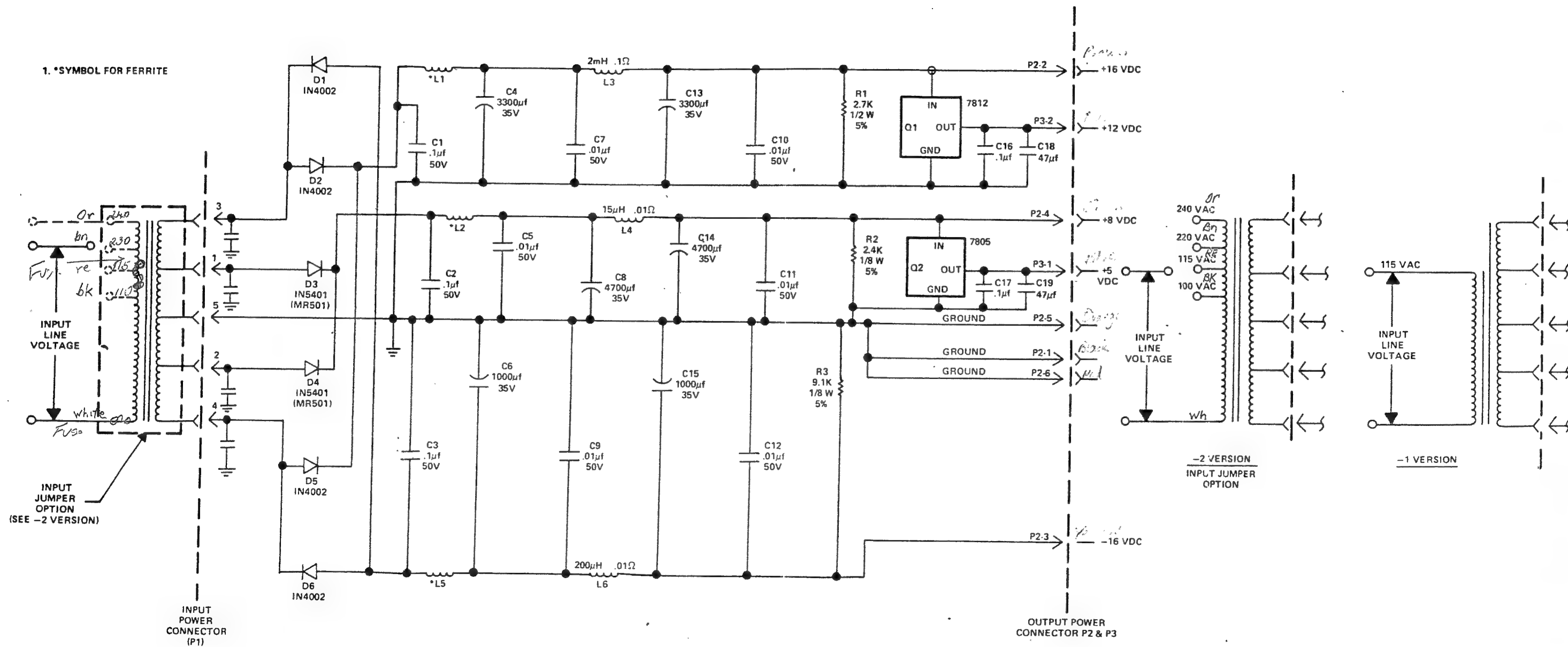


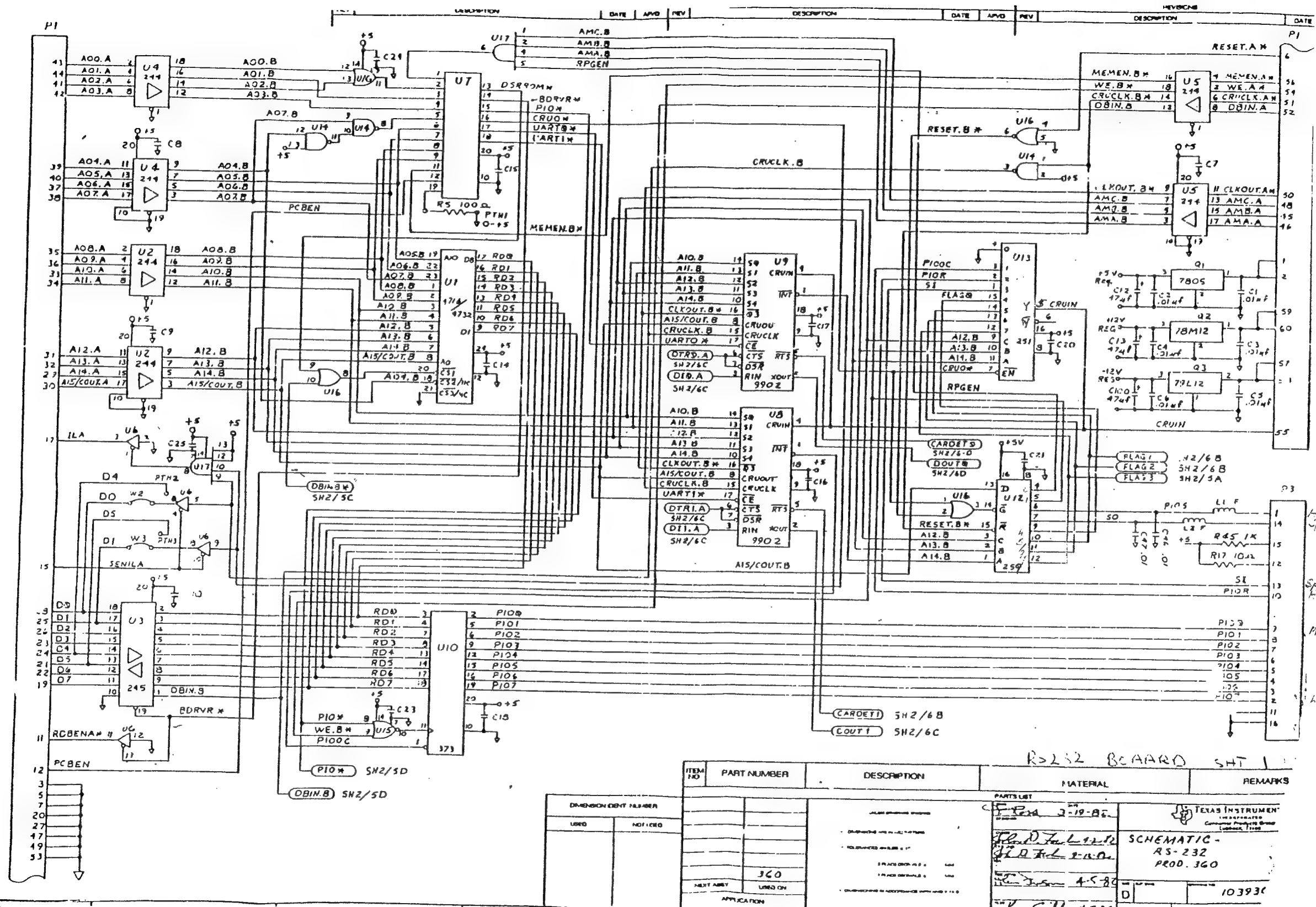






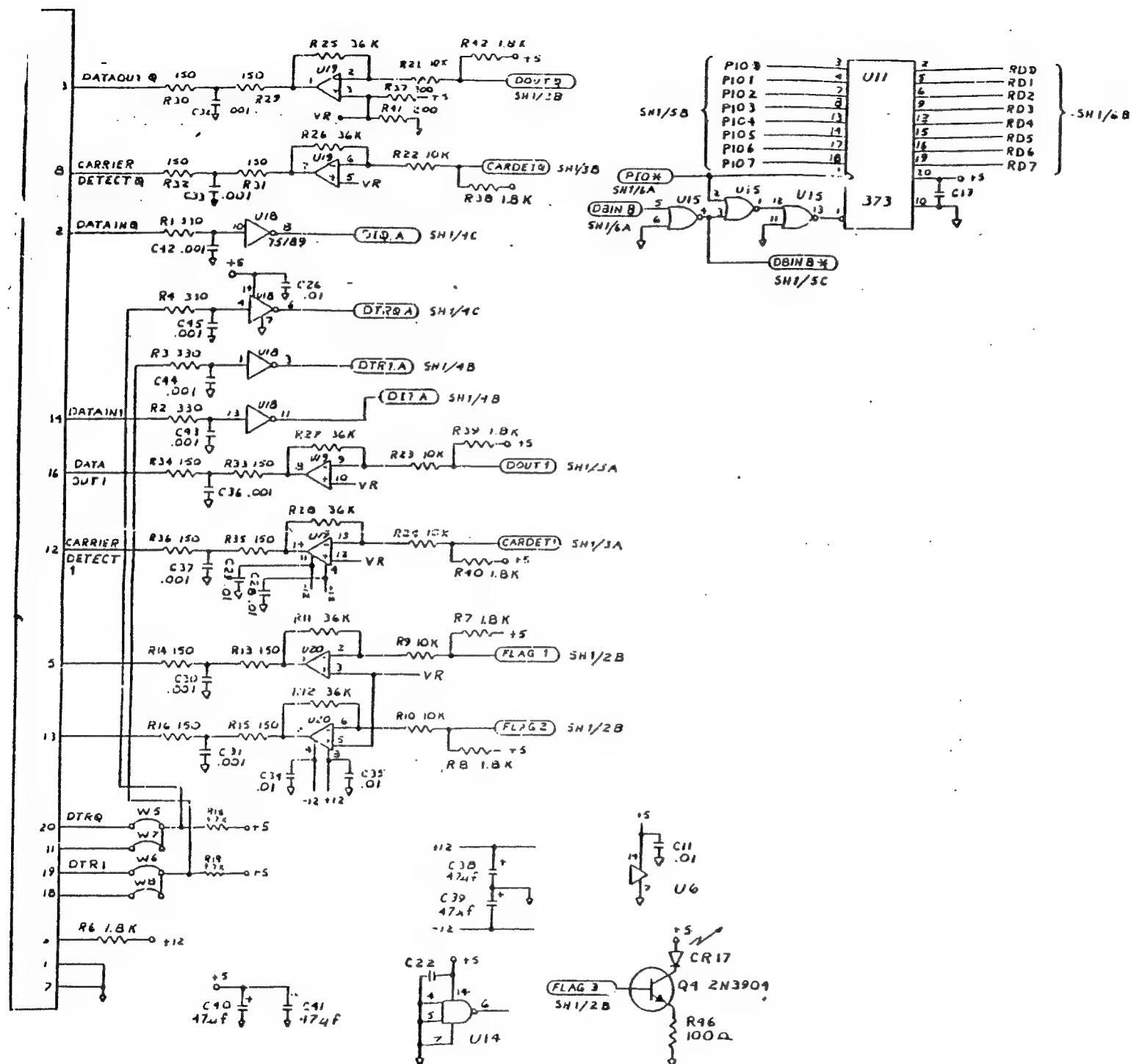






RS232 CARD SHT 1.

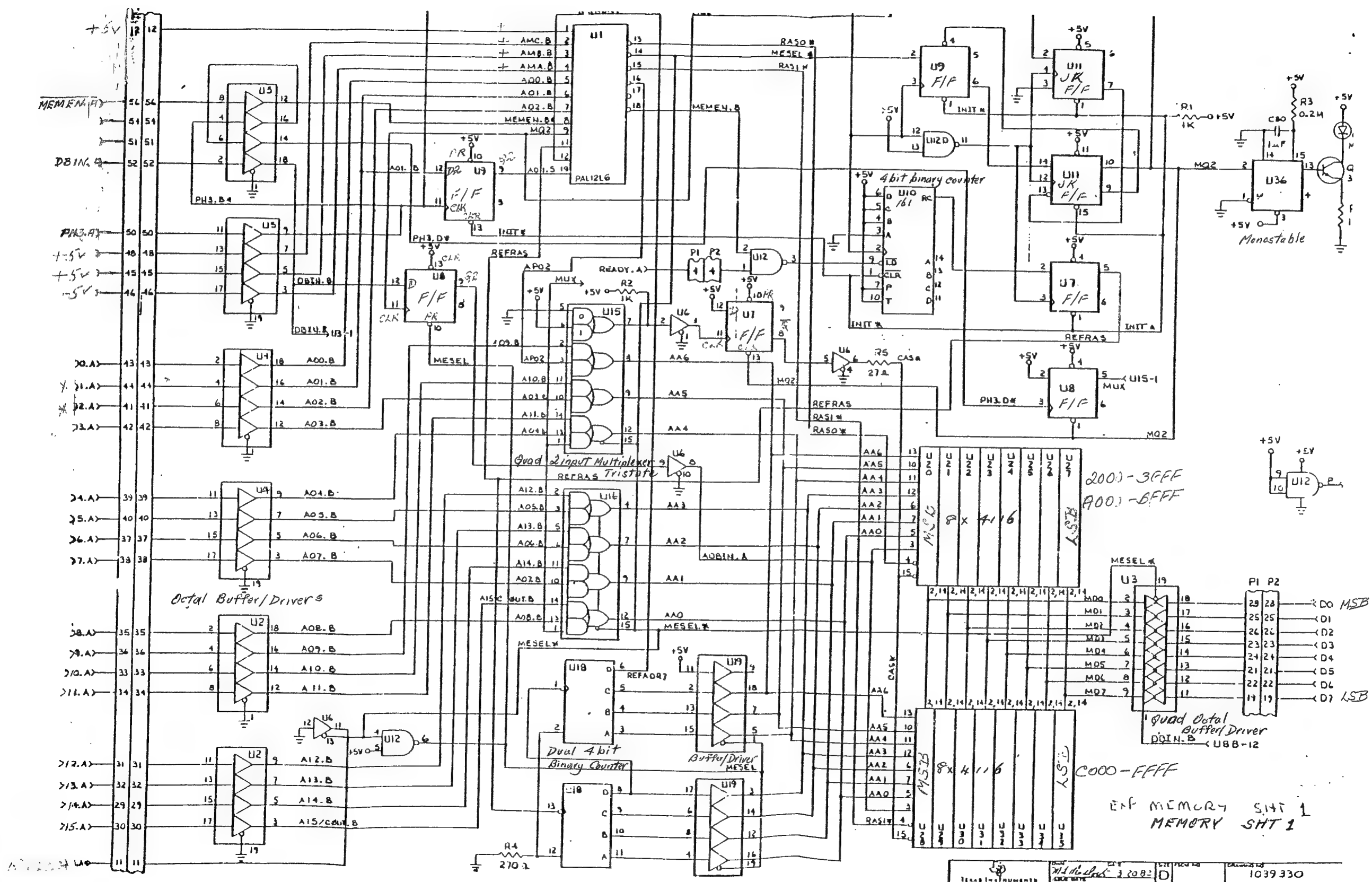


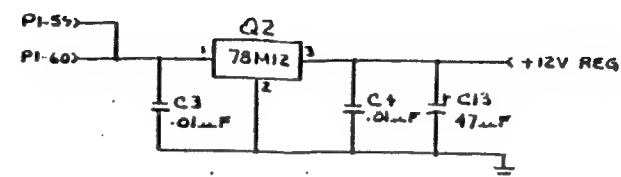


RS232 CARD SHTR  
RS232 BOARD SHTR

DIMENSIONS - IT NAMES	
USED	NOT USED

DESIGN NO.	1039308
DATE	3-19-82
DESIGNED BY	
CHECKED BY	
APPROVED BY	



[illegible]

EXPANSION MEMORY SHT2 Power

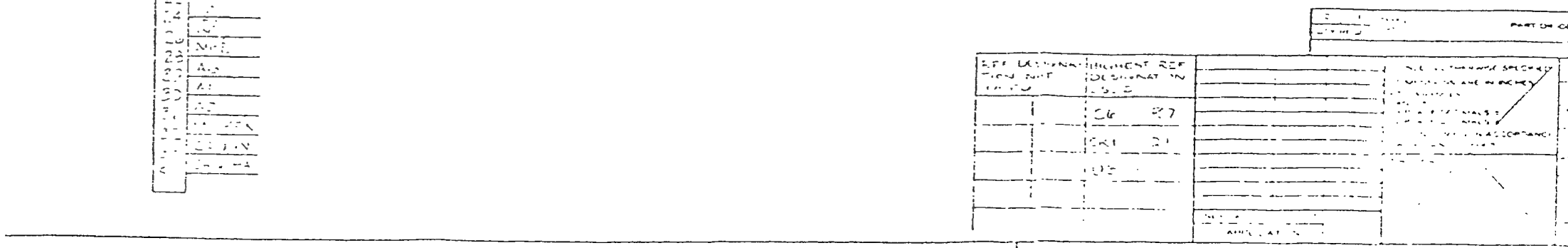




NOTE: ALL OF THE FIGURE BELOW IS 74LS SERIES  
TTL. ALL CAPS ARE 0.01 $\mu$ F.



A horizontal number line with tick marks at 3, 4, 5, 6, 7, and 8. A downward arrow points to the midpoint between 4 and 5.



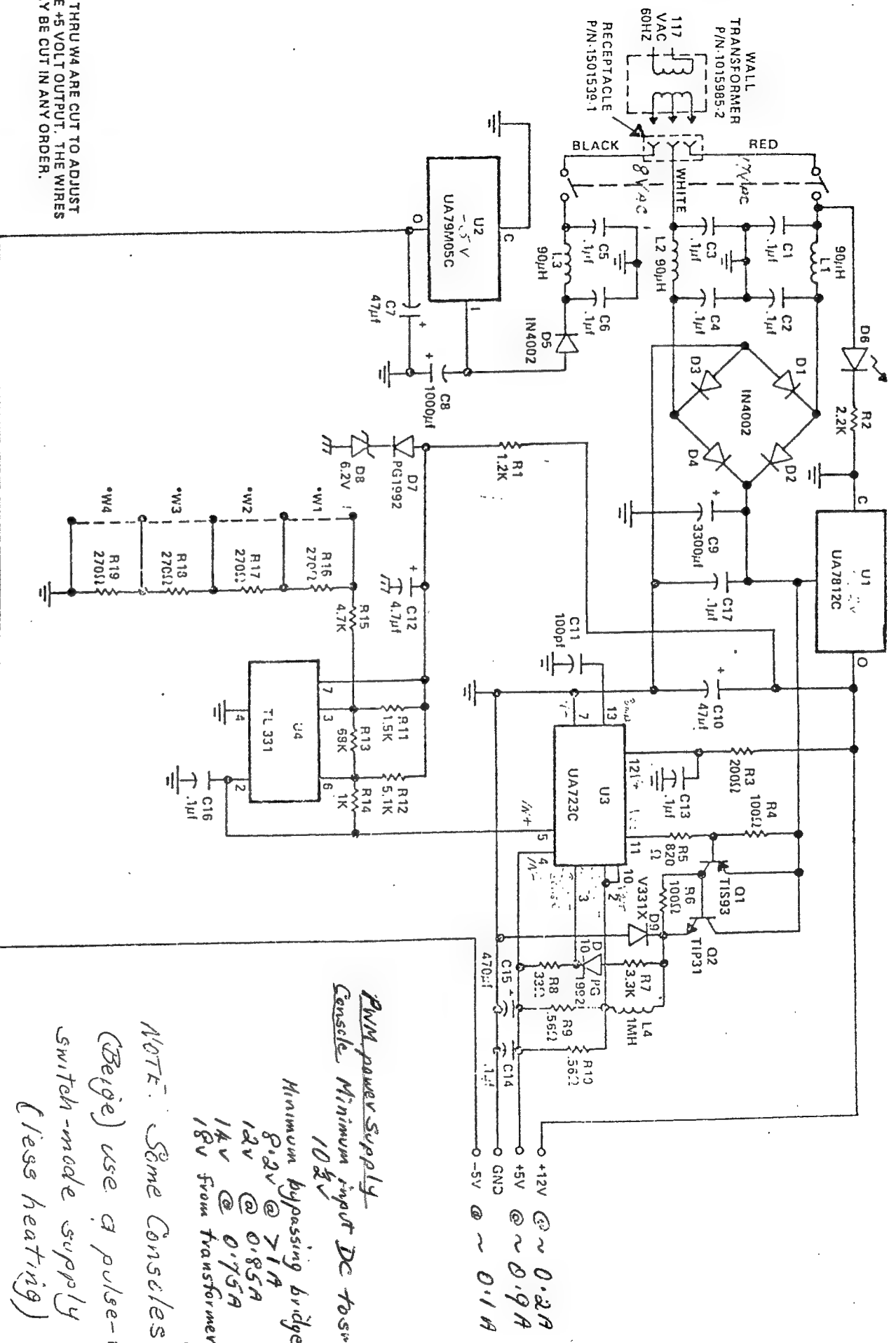
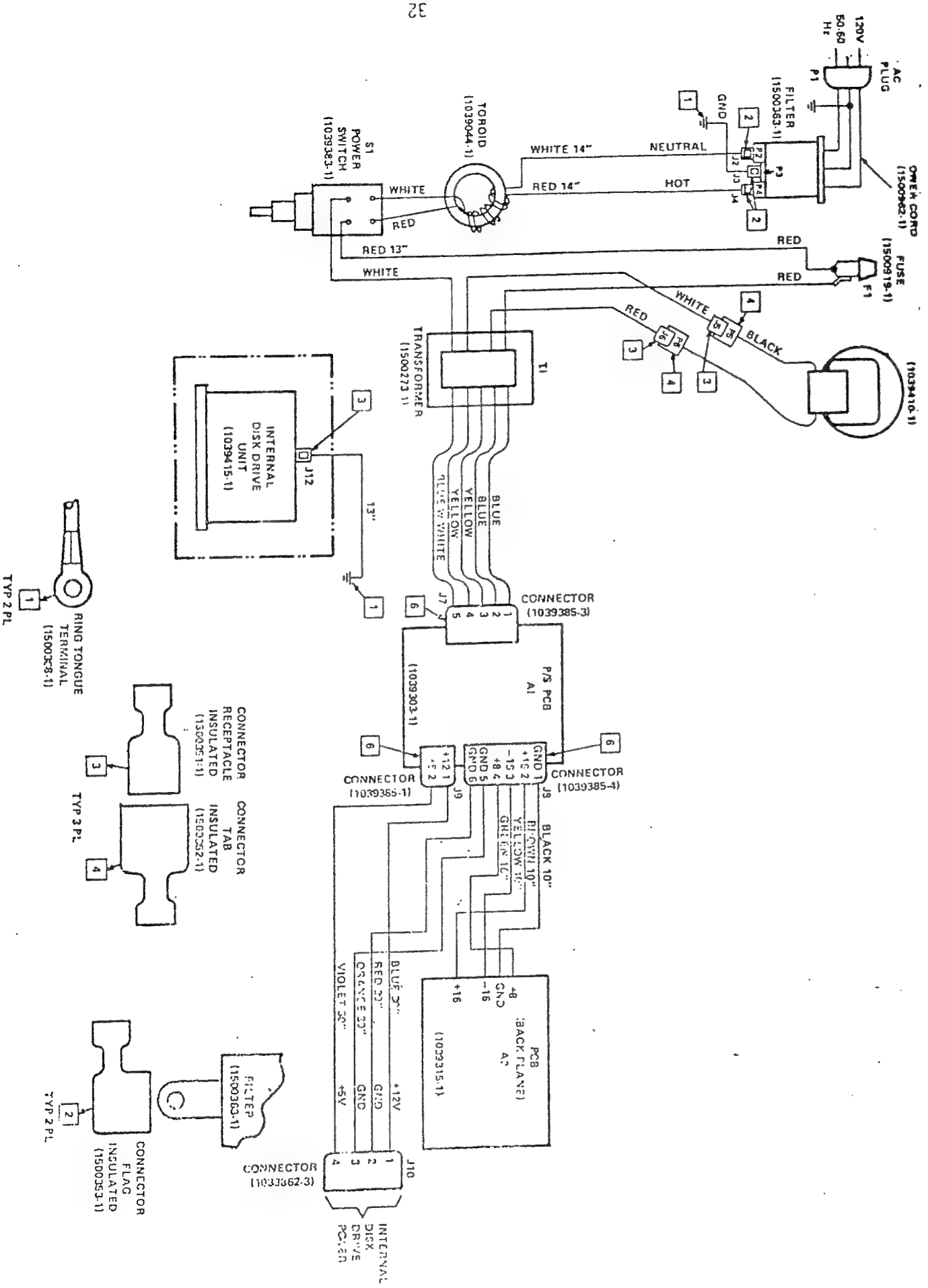
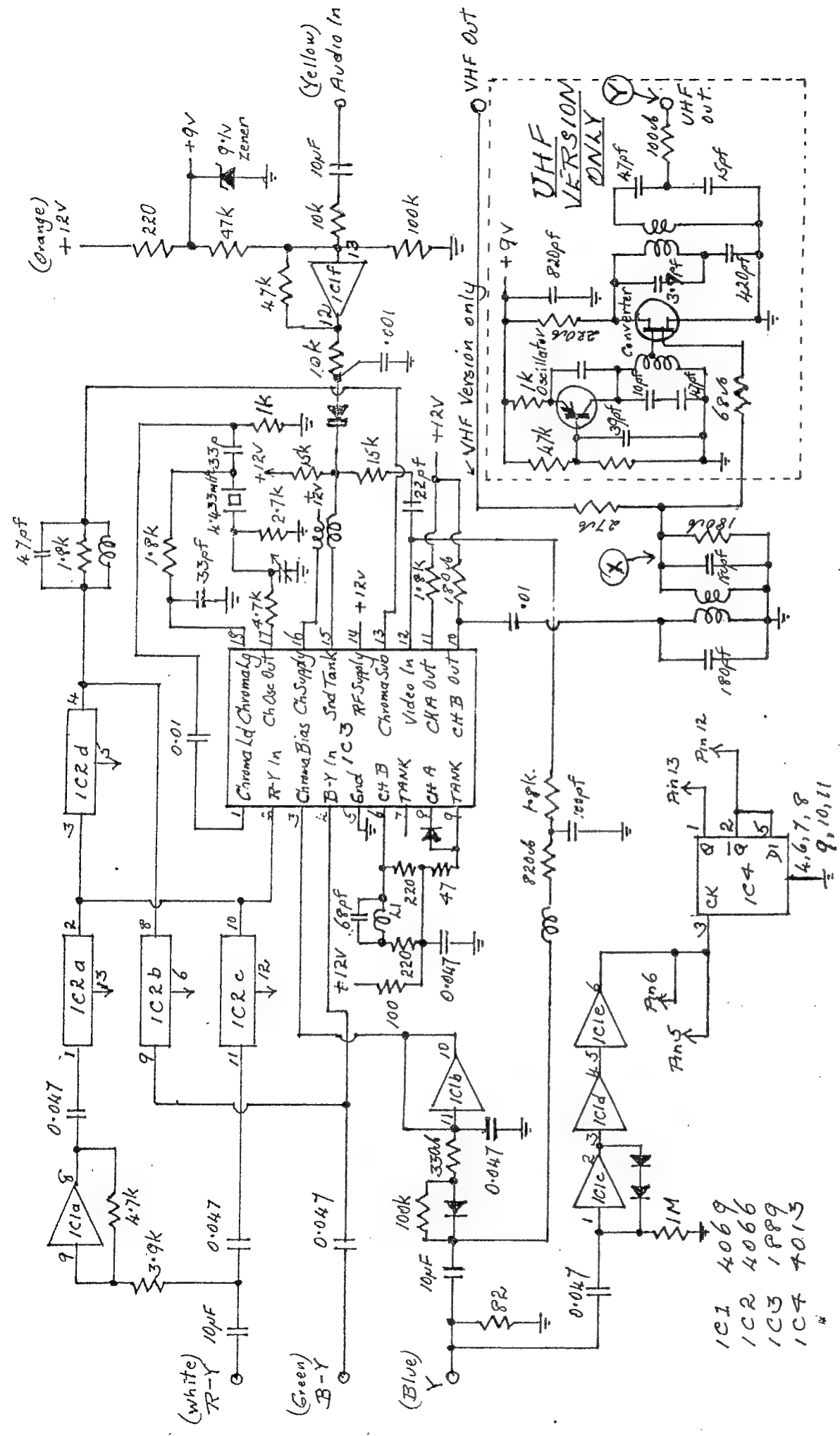


FIGURE G. T1-93/4A CONSOLE POWER SUPPLY

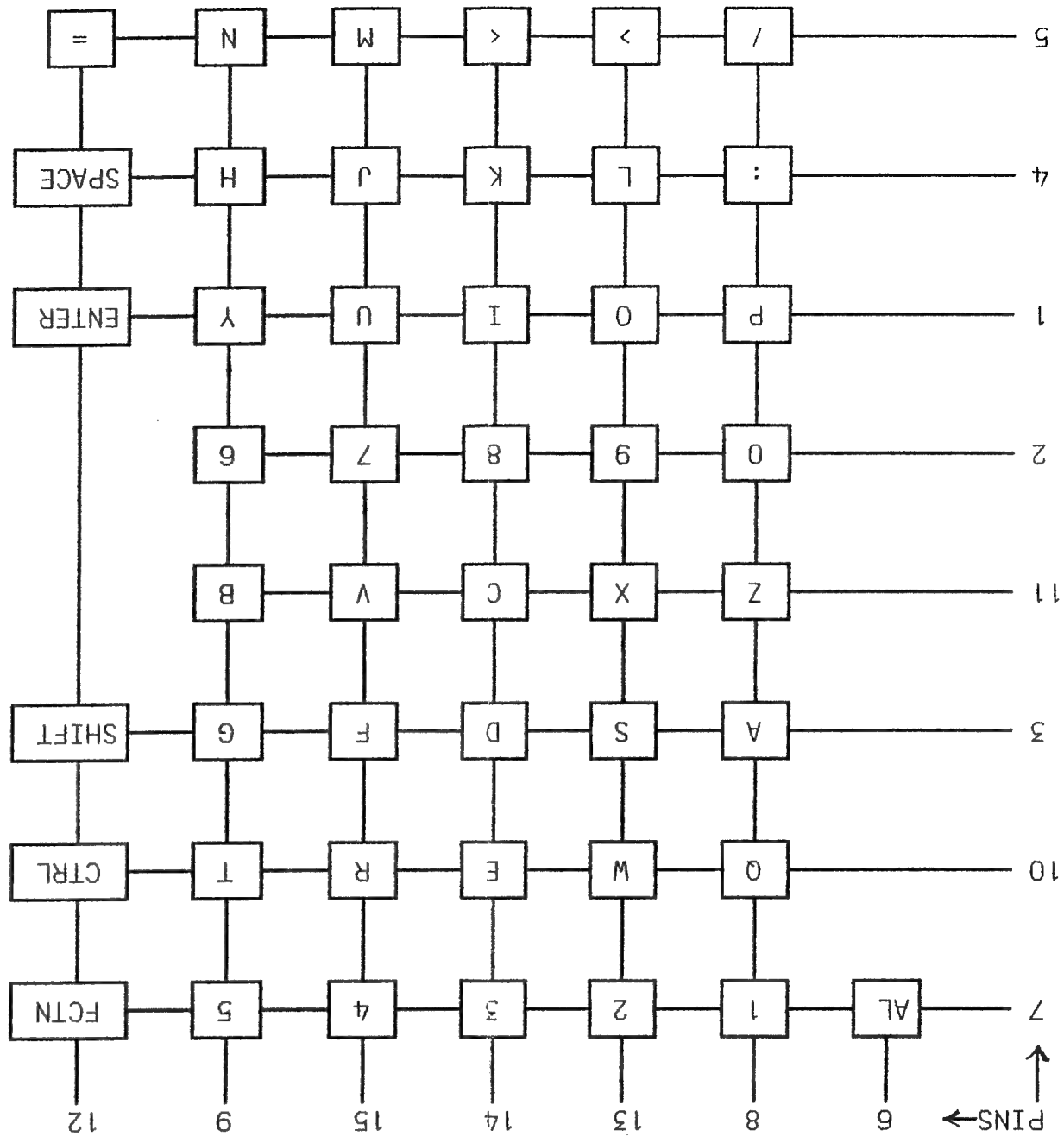


# MODULATOR CIRCUIT



PIN 1 IS NEAREST TO JOYSTICK PORT ON

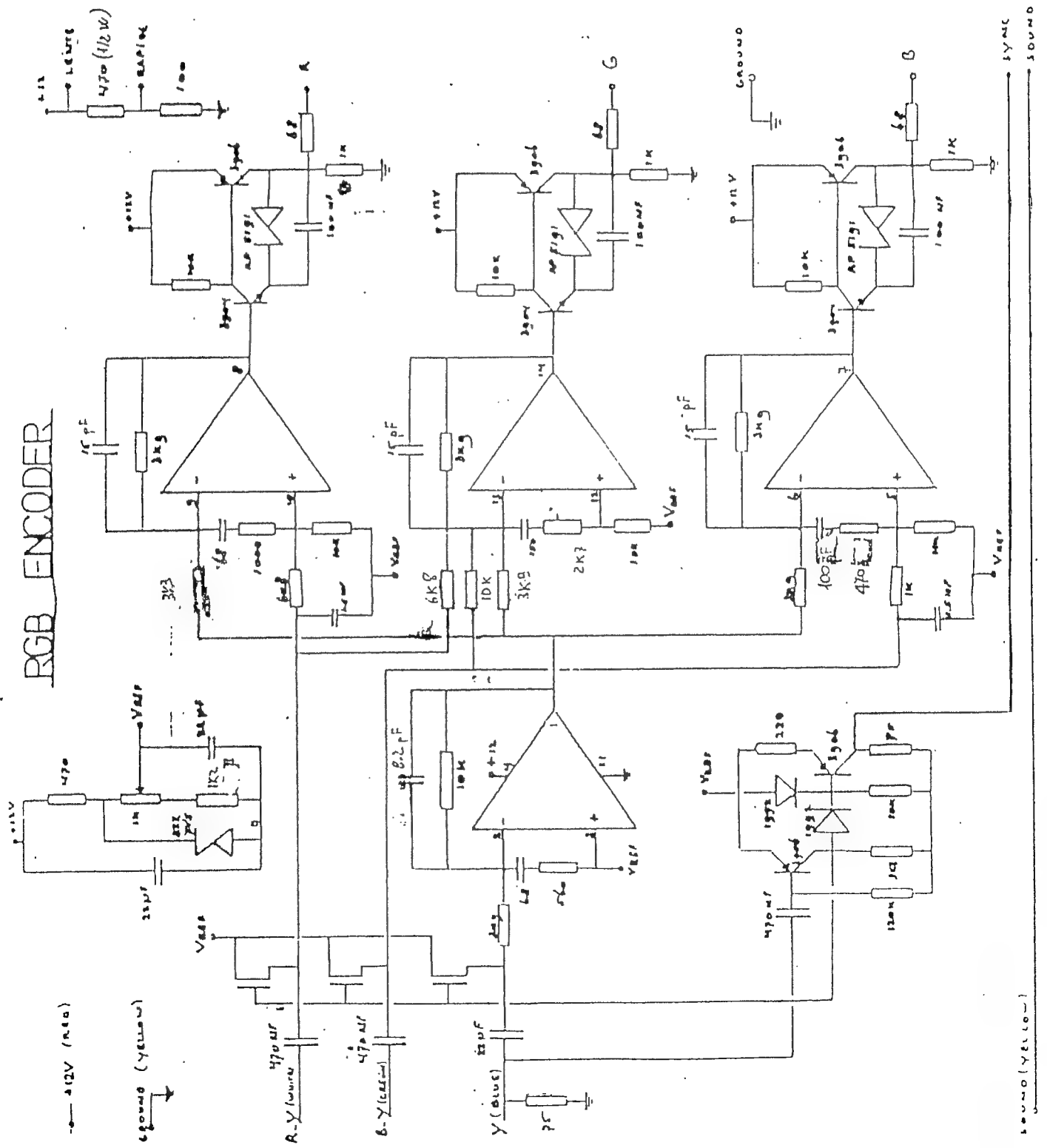
TI99/4A KEYBOARD  
WIRING MATRIX





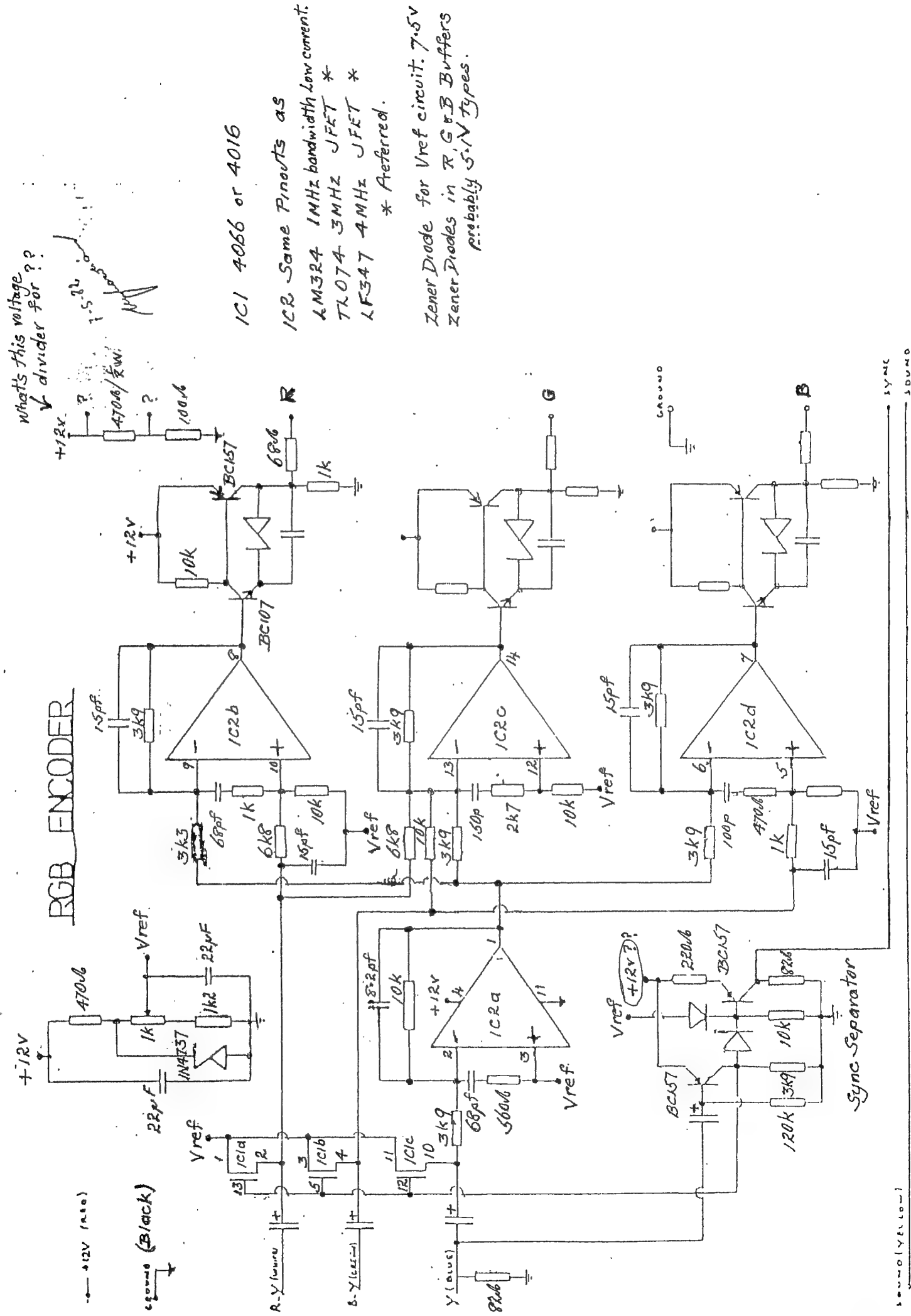
ORIGINAL SUPPLIED BY T.I.

# RGB ENCODER



Handwritten signature and date: 1-5-82

NOTE. Components & values Suggested only



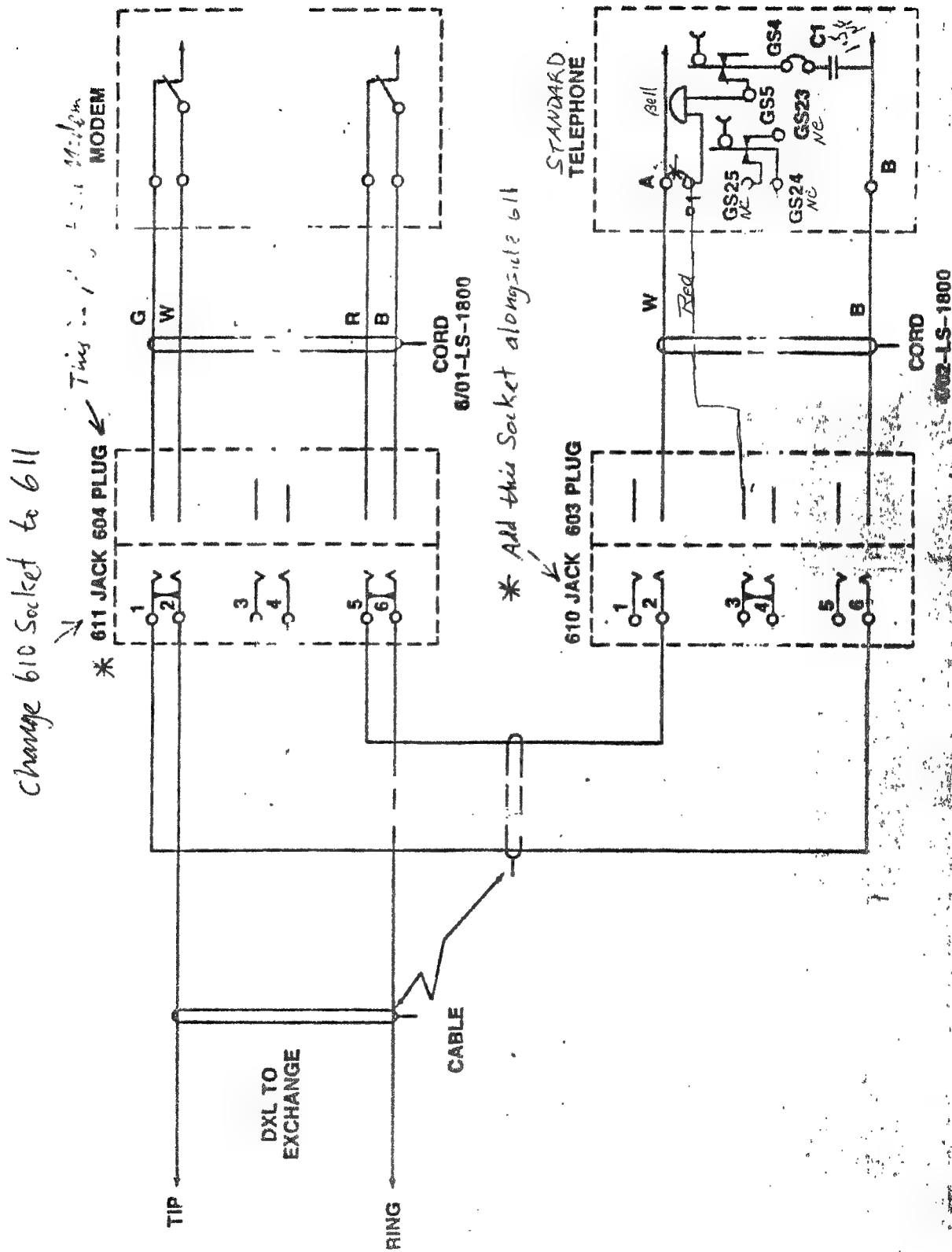
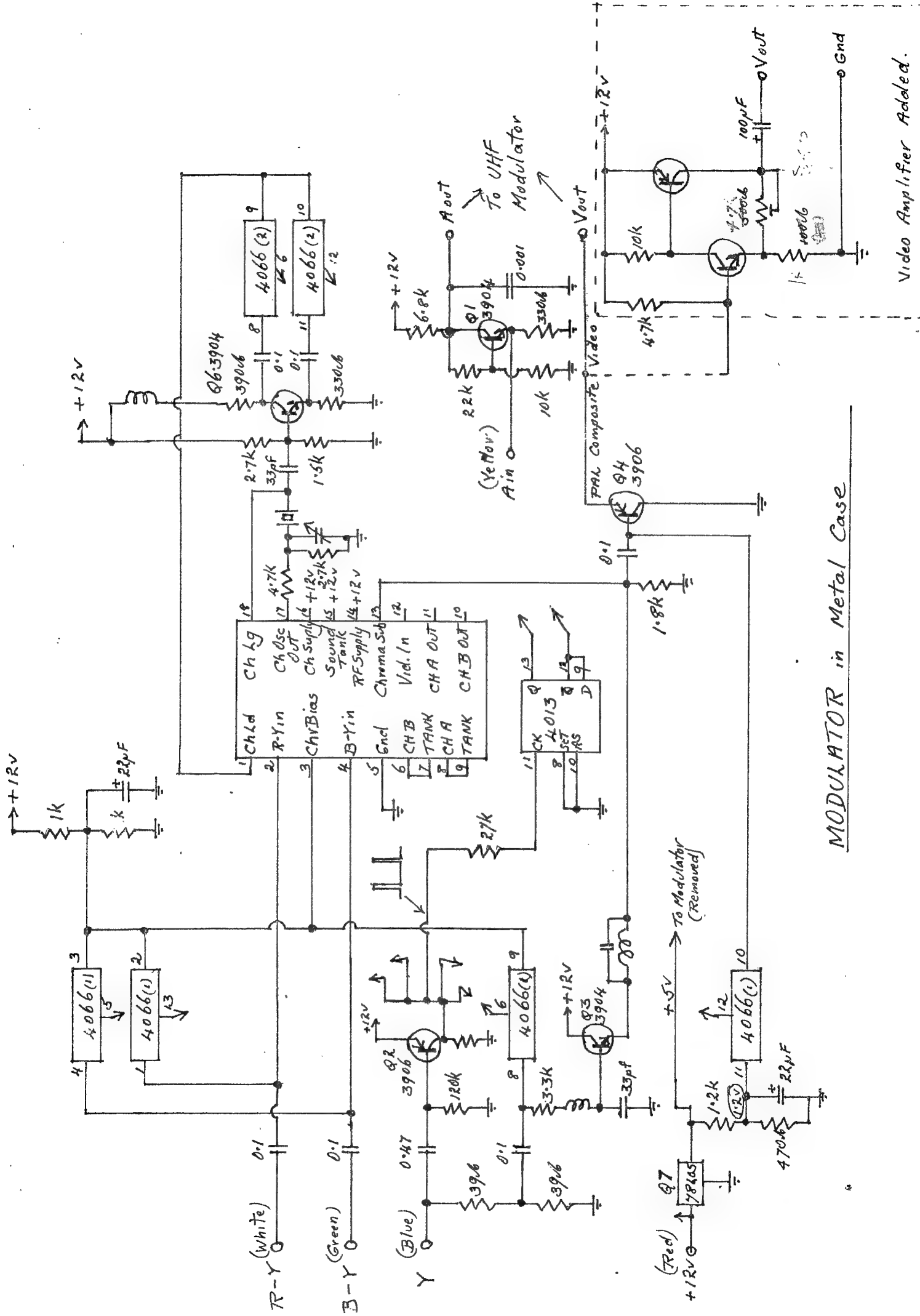


Figure 3-2. Connector Schematic

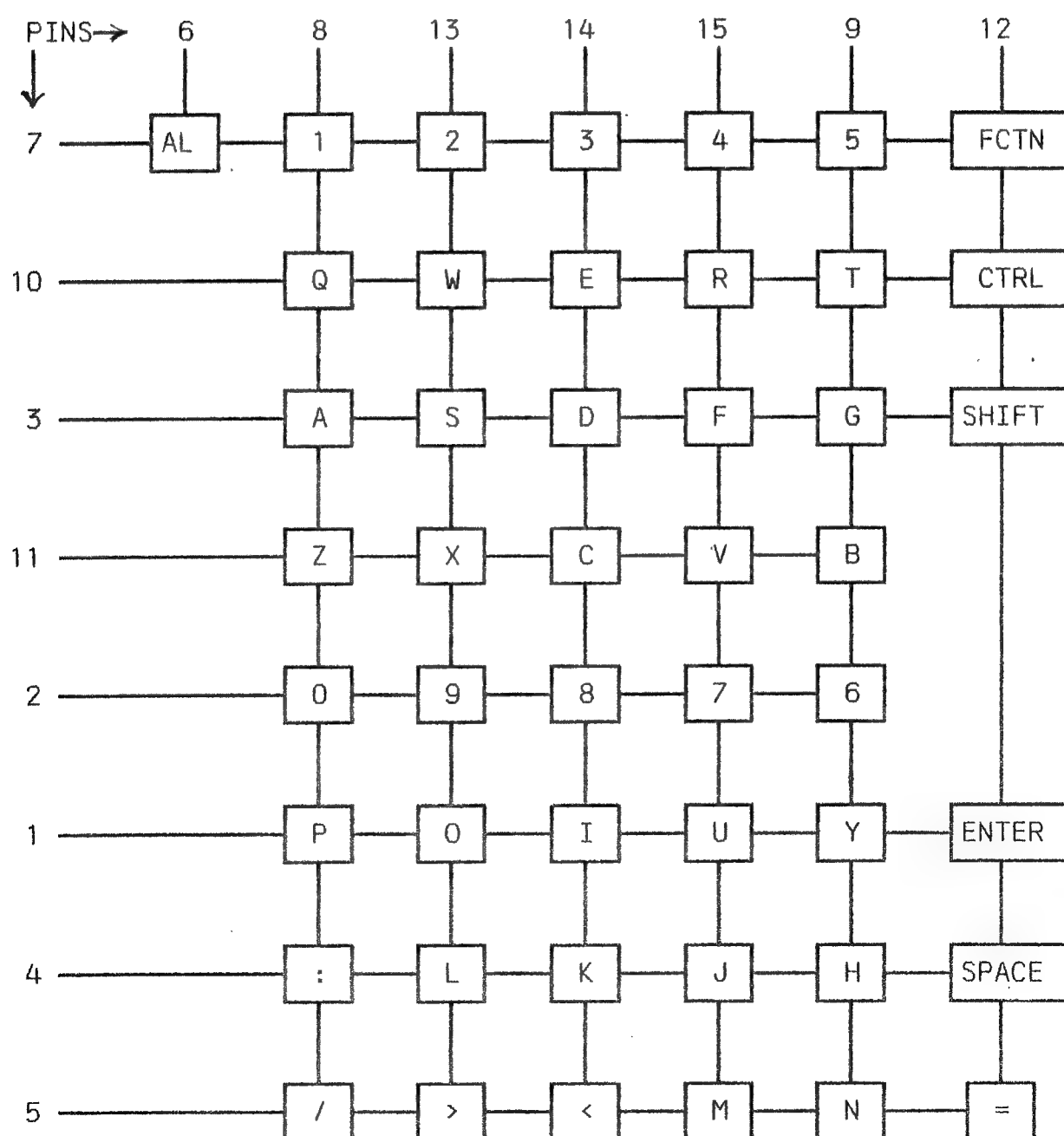


MODULATOR in Metal Case



# TI99/4A KEYBOARD WIRING MATRIX

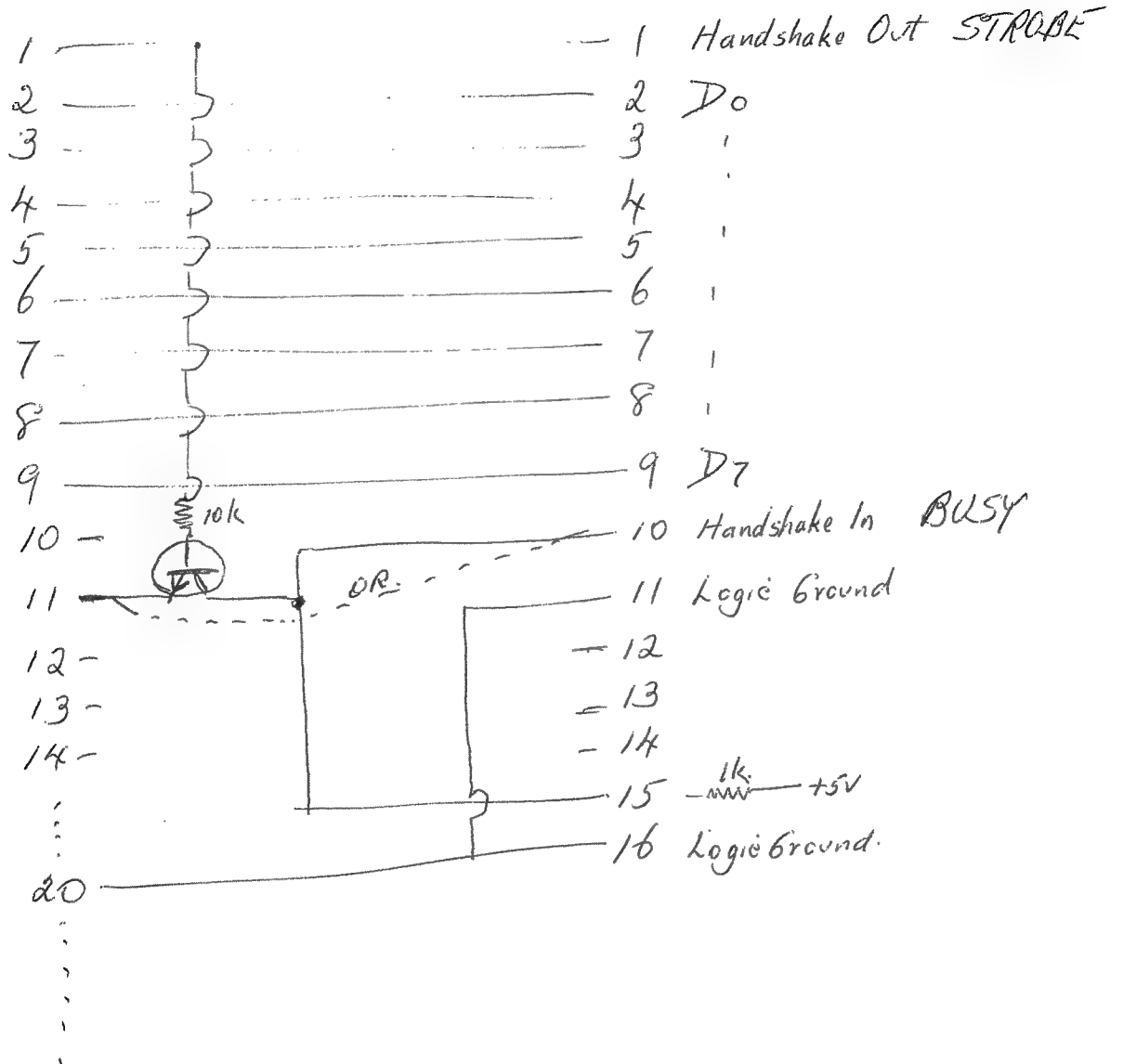
PIN 1 IS NEAREST TO JOYSTICK PORT ON  
THE MOTHERBOARD.



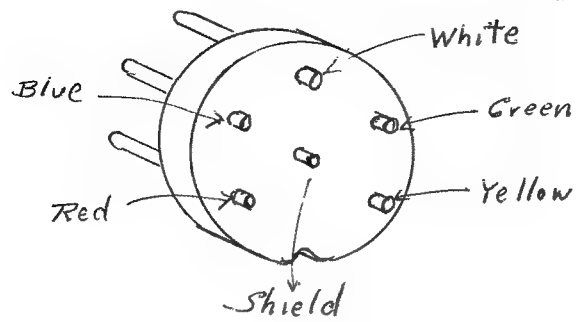
PRINTER CABLE

CENTRONICS

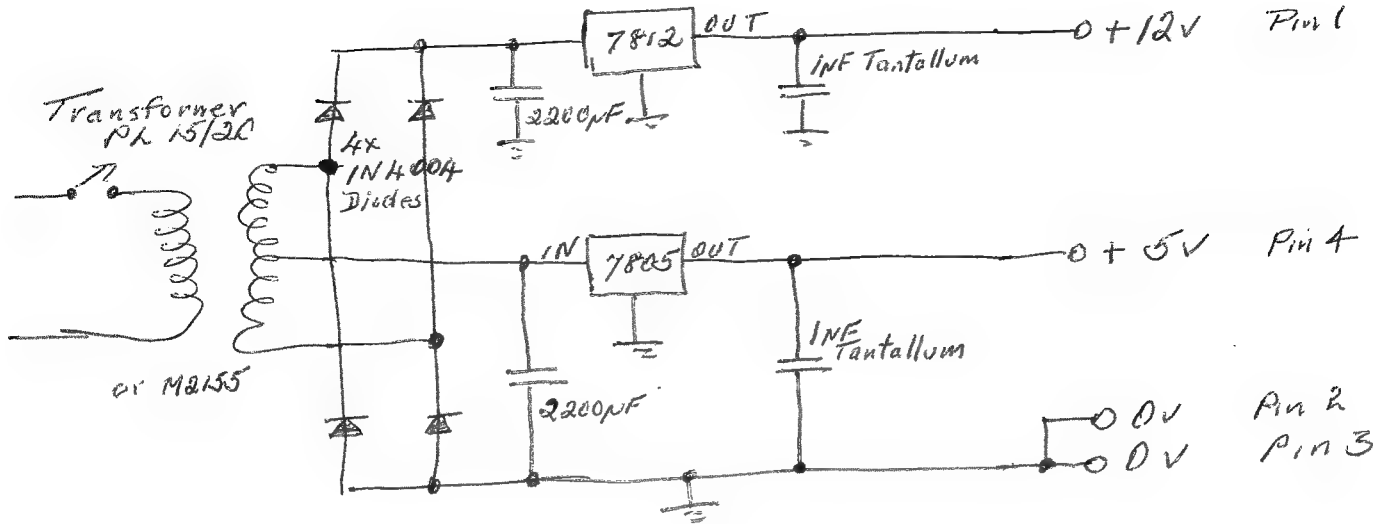
F10

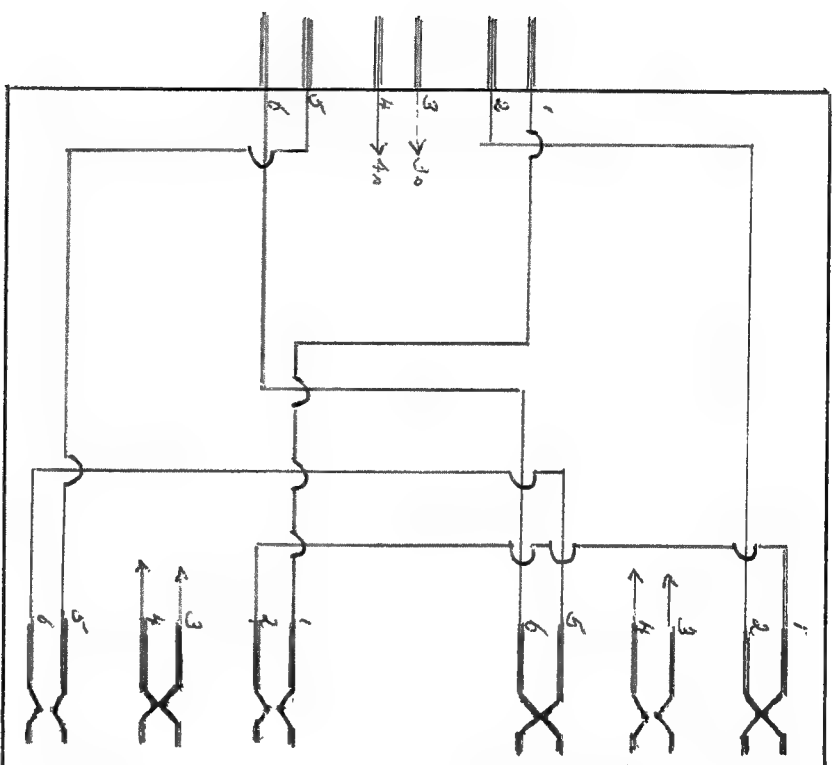


# Modulator Plug.- Wiring



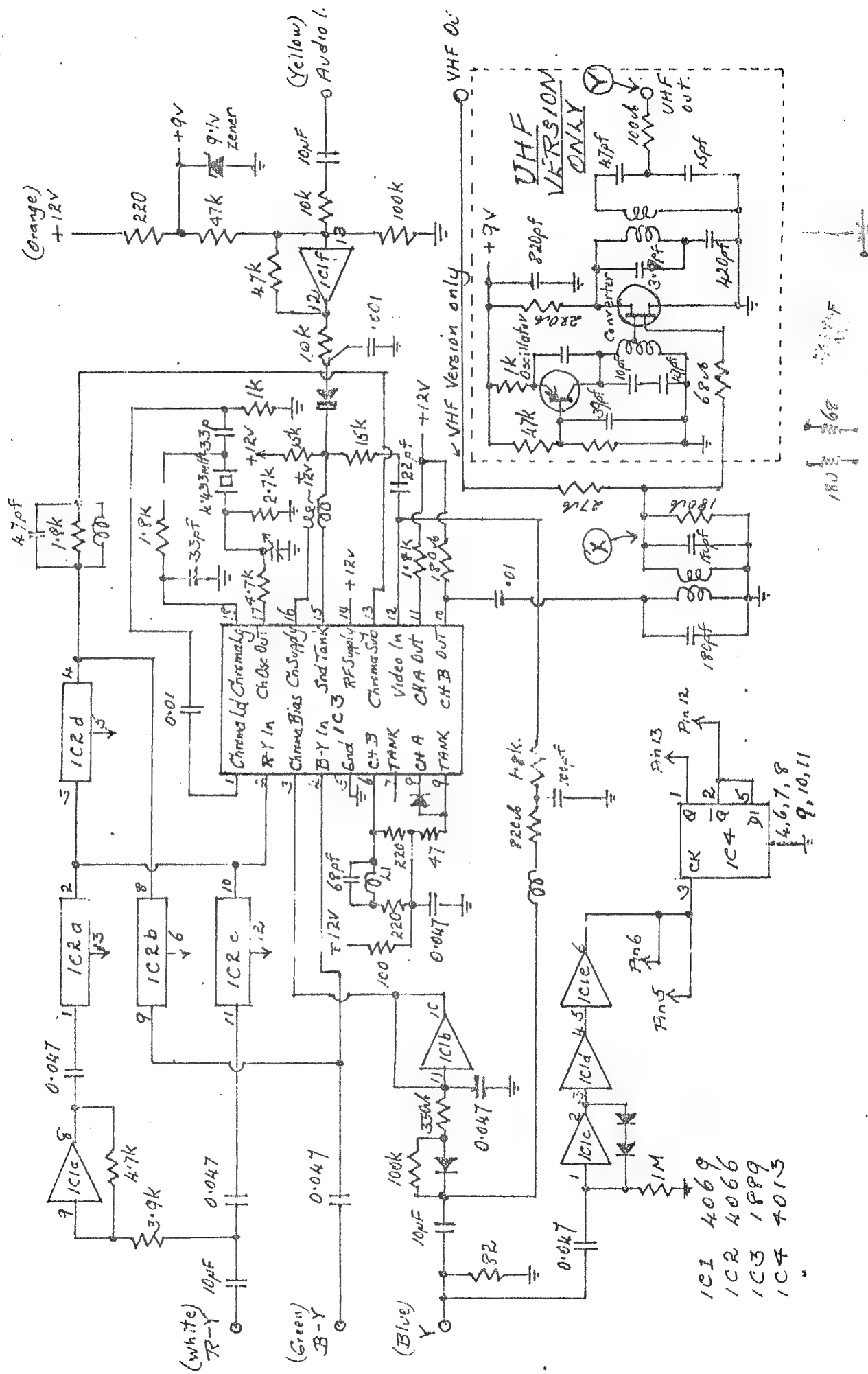
# DISK DRIVE POWER SUPPLY





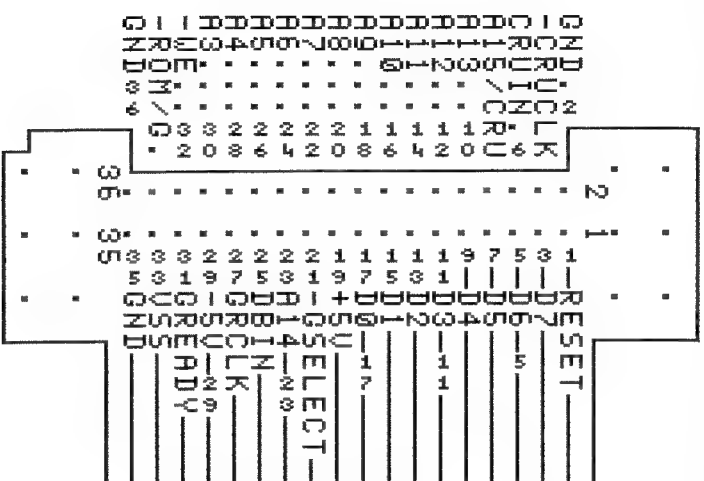
Modem

Phone

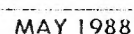




# FRONT PORT CONNECTIONS Rear View - Grom extender



by Joe Spiegel



# PRINTER AND CABLE CONNECTIONS

		PRINTER end.		TI end	
PRINTER	DIR	36PIN COLOUR	25PIN	16PIN	
1 STROBE	IN	1✓ BROWN	1	1✓ OUT	
2 DATA 0	IN	2✓ RED	2	2✓	
3 DATA 1	IN	3✓ ORANGE	3	3✓	
4 DATA 2	IN	4✓ PINK	4	4✓	
5 DATA 3	IN	5✓ YELLOW	5	5✓	
6 DATA 4	IN	6✓ GREEN	6	6✓	
7 DATA 5	IN	7✓ AQUA	7	7✓	
8 DATA 6	IN	8✓ BLUE	8	8✓	
9 DATA 7	IN	9✓ L BLUE	9	9✓	
10 ACK	OUT	10 PURPLE	10		
11 BUSY	OUT	11✓ GREY	11	10✓ IN	
12 PAP OUT	OUT	12 WHITE	12		
13 SELECT	OUT	13 BLACK	13		
14 N/C		14 BN/WH	14		
15 N/5					
16 SIG GND					
17 CHAS "					
18 +5V DC					
*****					
19 GND		19✓ GN/BK	19	11✓ 16✓	
20 GND		20 N/C			
21 GND		21 AQ/BK	20		
22 GND		22 N/C			
23 GND		23 BL/WH	21		
24 GND		24 N/C			
25 GND		25 L BL/WH	22		
26 GND		26 N/C			
27 GND		27 PU/WH	23		
28 GND		28 N/C			
29 GND		29 GY/BK	24		
30 GND		30 WH/BK	25		
31 RESET	IN	31 OR/BK	16		
32 ERROR	IN	32 RE/BK	15		
33 EXT GND		33 YE/BK	18		
34 N/C		34 N/C			
35 N/C		35 N/C			
36 SELC-IN	IN	36 PK/BK	17		

## Q&amp;A

Over the past few months we have received some questions regarding the Power Up routine and its sequence of events.

The Power Up routine resides in console ROM and console GROM chip 0, with the majority of it in GROM. Listed below is the sequence of events, as we know it, that take place from the moment power is applied to the console.

1. When power is first applied to the 9900 microprocessor it executes a level 0 interrupt which is a reset. When this happens it knows to grab its workspace pointer from address >0000 and its program counter or instruction pointer from address >0002. In the 99/4A these addresses are burned into the console ROM. Once it has these addresses it begins to execute the code that is pointed to by address >0002.

2. The code pointed to by address >0002 loads R13 with >9800, the GROM read address, R14 with >0100, the status flag and R15 with >8C02, the VDP write address.

3. The balance of the Power Up routine is in GROM chip 0. First it clears out the sound list indicator at >83CE and then it turns off the Speech Synthesizer, if you have one, and turns off the sound generators.

4. Next it initializes the Data and Subroutine GPL stacks in Scratch Pad RAM.

5. Then it loads the VDP registers with default values stored in a GROM table.

6. Most of the Scratch Pad RAM is then cleared out by writing zeros to it.

7. The keyboard interrupts are then disabled and the cassette audio gate is turned on, this allows you to hear the cassette through your monitor speaker.

8. Next the VDP and external interrupts are enabled.

9. The cassette motors are then enabled, this sets up the remote jack so the motors can run.

10. Next it generates a **BEEP** sound by CALLing the TON1 routine in GROM chip 0.

11. And then it determines the size of VDP RAM, 4K or 16K, and sets bit 0 in VDP register 1 to 1 for 16K. I guess that TI thought they might produce a 4K console at one point in time.

12. The first 4K of VDP RAM is then cleared out and the default color and character tables are moved from GROM to VDP RAM.

13. Next the keyboards are initialized by scanning them, ie: CALL KEY(5.. ,CALL KEY(4.. , CALL KEY(3.. etc.

14. Then it moves the data for the Title Screen into the screen image table. At this point the screen can not be seen because it is turned off. Bit 1 in VDP register 1 is set to zero.

15. The ROMs(DSR) and GROMs are then searched for power up headers and if they have one they are executed.

16. After the search and execution of power up headers is complete the Title Screen is displayed by writing a 1 to bit 1 of VDP register 1. This is the first time that you are able to see the Title Screen.

17. Next it initializes the Random Number Generator and then waits for you to 'PRESS ANY KEY'. After you press a key it generates another BEEP sound.

18. And then it searches through the cartridge ROM and GROMs for application programs and builds a list of their names. ie: Editor/Assembler, Extended Basic, Basic etc. Note: some consoles, V2.0 (C) 1983, do not search ROM only GROM.

19. At this point the screen is once again turned off, VDP register 1, bit 1 is set to zero and then it sets up the menu screen with the application programs it found.

20. After the menu is set up it turns on the screen by writing a 1 to bit 1 of VDP register 1, and waits for you to make a selection.

21. If you make an illegal selection it generates a HONK sound, otherwise it sets up the starting address for the application program you selected and branches to it.

## Q&A

Over the past few months we have received some questions regarding the Power Up routine and its sequence of events.

The Power Up routine resides in console ROM and console GROM chip 0, with the majority of it in GROM. Listed below is the sequence of events, as we know it, that take place from the moment power is applied to the console.

1. When power is first applied to the 9900 microprocessor it executes a level 0 interrupt which is a reset. When this happens it knows to grab its workspace pointer from address >0000 and its program counter or instruction pointer from address >0002. In the 99/4A these addresses are burned into the console ROM. Once it has these addresses it begins to execute the code that is pointed to by address >0002.
2. The code pointed to by address >0002 loads R13 with >9800, the GROM read address, R14 with >0100, the status flag and R15 with >8C02, the VDP write address.
3. The balance of the Power Up routine is in GROM chip 0. First it clears out the sound list indicator at >83CE and then it turns off the Speech Synthesizer, if you have one, and turns off the sound generators.
4. Next it initializes the Data and Subroutine GPL stacks in Scratch Pad RAM.
5. Then it loads the VDP registers with default values stored in a GROM table.
6. Most of the Scratch Pad RAM is then cleared out by writing zeros to it.
7. The keyboard interrupts are then disabled and the cassette audio gate is turned on, this allows you to hear the cassette through your monitor speaker.
8. Next the VDP and external interrupts are enabled.
9. The cassette motors are then enabled, this sets up the remote jack so the motors can run.
10. Next it generates a BEEP sound by calling the TONI routine in GROM chip 0.
11. And then it determines the size of VDP RAM, 4K or 16K, and sets bit 0 in VDP register 1 to 1 for 16K. I guess that TI thought they might produce a 4K console at one point in time.
12. The first 4K of VDP RAM is then cleared out and the default color and character tables are moved from GROM to VDP RAM.
13. Next the keyboards are initialized by scanning them. ie: CALL KEY(5.. ,CALL KEY(4.. , CALL KEY(3.. etc.
14. Then it moves the data for the Title Screen into the screen image table. At this point the screen can not be seen because it is turned off. Bit 1 in VDP register 1 is set to zero.
15. The ROMs(DSR) and GROMs are then searched for power up headers and if they have one they are executed.
16. After the search and execution of power up headers is complete the Title Screen is displayed by writing a 1 to bit 1 of VDP register 1. This is the first time that you are able to see the Title Screen.
17. Next it initializes the Random Number Generator and then waits for you to 'PRESS ANY KEY'. After you press a key it generates another BEEP sound.
18. And then it searches through the cartridge ROM and GROM for application programs and builds a list of their names. ie: Editor/Assembler, Extended Basic, Basic etc. Note: some consoles, V2.0 (C) 1983, do not search ROM only GROM.
19. At this point the screen is once again turned off, VDP register 1, bit 1 is set to zero and then it sets up the menu screen with the application programs it found.
20. After the menu is set up it turns on the screen by writing a 1 to bit 1 of VDP register 1, and waits for you to make a selection.
21. If you make an illegal selection it generates a HONK sound, otherwise it sets up the starting address for the application program you selected and branches to it.